インストールマニュアル

動画 DB システム

発行日 2020 年 3 月 31 日
公立大学法人会津大学
株式会社東日本計算センター

目　次

1．本書について 1
　1.1．インストール環境 1
2．導入準備 2
3．Java の導入 2
　3.1．JRE 2
　3.2．JDK 2
　3.3．Java のバージョン切り替え 3
4．Apache2 の導入 4
　4.1．インストール 4
　4.2．設定 4
　　4.2.1．security.conf 4
　　4.2.2．dir.conf 5
　　4.2.3．apache2.conf 5
　　4.2.4．再起動 5
5．Tomcat の導入 6
　5.1．準備 6
　5.2．インストール 6
　5.3．サービスファイル設定 6
　5.4．起動 7
6．Apache と Tomcat の連携 8
　6.1．プロキシ機能の有効化 8
　6.2．仮想ホストの設定 8
　6.3．Apache の再起動 9
7．FFmpeg の導入 10
　7.1．準備 10
　7.2．Openh264 インストール 10
　7.3．FFmpeg インストール 10
　7.4．設定 11
　7.5．起動確認 11
8．PostgreSQL の導入 12
　8.1．インストール 12
　8.2．設定 13
　　8.2.1．ロール作成 13
　　8.2.2．DB 作成 13

8.2.3. テーブルとシーケンス作成 13
8.2.4. ユーザー登録 14
8.2.5. オーナー変更 14
9. Spring の導入 15
9.1. 準備 15
9.2. ダウンロード 15
9.3. 日本語化 15
9.4. Maven 17
9.4.1. インストール 17
9.4.2. 動作確認 17
9.5. 設定 18
9.5.1. Maven 18
9.5.1.1. インストール 18
9.5.1.2. ユーザー設定 20
9.6. プロジェクトファイルのインポート 22
9.7. プロジェクトの更新 23
9.8. database.properties の設定 24
9.9. War ファイルの生成 25
9.9.1. ターゲットディレクトリのクリーン 25
9.9.2. War ファイルの生成 25
10. Web アプリケーションの設定 27
10.1. Web アプリケーション用ディレクトリ作成 27
10.2. ディレクトリ所有者設定 28
10.3. Tomcat シンボリックリンク参照有効化 28
11. Web アプリのデプロイ 29
11.1. war ファイルの設定 29
11.2. Tomcat サービス停止 29
11.3. シンボリックリンクの作成 29
11.4. 共有ライブラリの設定 30
11.5. Tomcat サービス開始 30

1．本書について

本書は動画 DB システムの動作環境の構築手順を記載します。本システムの操作に関しては「ユーザーズマニュアル_動画 DB システム」を参照ください。

1.1．インストール環境

本書では以下の環境を想定しています。

表 1-1 動作環境一覧

環境		バージョン	補足
OS	Ubuntu	16.04 LTS	-
CPU	Core i7 相当以上	-	-
メモリ	8GB 以上推奨	-	-
ストレージ	100GB 以上推奨	-	-
開発言語	Java	9	-
ミドルウェア	Apache	2.4.18	Apache License 2.0
	Tomcat	9.0.4	Apache License 2.0
	PostgreSQL	11.4	PostgreSQL License
フレームワーク	Spring	3.9.9	Eclipse Public License ver1.0
依存ライブラリ	FFmpeg	3.4.7	LGPL ライセンス
	OpenJDK	9.0.4	GPL ライセンス
	JDBC Driver	42.2.0	BSD

2．導入準備

各種ライブラリ等の導入準備をします。導入するバージョンは本システムの環境構築時点となります。ライブラリによっては最新版もしくは任意の世代までの公開となっている場合があります。その際は最新版を導入してください。

パッケージを更新します。

```
$ sudo apt update
```

ソースコードのダウンロードを行うために、Git と curl をインストールします。

```
$ sudo apt install git curl
```

3．Java の導入

3.1．JRE

Java アプリケーションの実行に必要な JRE(Java Runtime Environment)をインストールします。

```
$ sudo apt install default-jre
```

3.2．JDK

(1) Java アプリケーション開発に必要な JDK(Java Development Kit)をインストールします。まずは OpenJDK をダウンロードし解凍します。

```
$ cd /opt
$ sudo curl -OL https://download.java.net/java/GA/jdk9/9.0.4/binaries/openjdk-9.0.4_linux-x64_bin.tar.gz
$ sudo tar xzvf /opt/openjdk-9.0.4_linux-x64_bin.tar.gz
```

(2) OpenJDK の所有者を変更します。

```
$ sudo chown -R root:root /opt/jdk-9.0.4
```

(3) OpenJDK のシンボリックリンクの作成・登録を実施します。

```
$ sudo update-alternatives --install /usr/bin/java java /opt/jdk-9.0.4/bin/java 1
$ sudo update-alternatives --install /usr/bin/jar jar /opt/jdk-9.0.4/bin/jar 1
$ sudo update-alternatives --install /usr/bin/javac javac /opt/jdk-9.0.4/bin/javac 1
```

(4) OpenJDK のインストール完了確認を行います。以下のようにインストールした情報が出力されます。

```
$ java -version
openjdk version "9.0.4"
OpenJDK Runtime Environment (build 9.0.4+11)
OpenJDK 64-Bit Server VM (build 9.0.4+11, mixed mode)
```

3.3. Java のバージョン切り替え

今回は 9.0.4 を導入しましたが、複数のバージョンが共存しており、9.0.4 がアクティブでない場合は、図 3-1 のように切り替えることが可能です。

```
$ sudo update-alternatives --config java
```

図 3-1 Java のバージョン切り替え

4．Apache2 の導入

4.1．インストール

(1) Apache2 をインストールします。

```
$ sudo apt install apache2
```

(2) インストール完了確認を行います。以下のようにインストールした情報が出力されます。

```
$ apache2 -v
Server version: Apache/2.4.18 (Ubuntu)
Server built:     2019-10-08T13:31:25
```

4.2．設定

4.2.1．security.conf

クライアントに不要なサーバー情報を公開しないように設定を変更します。普段使い慣れたエディタで編集ください。本書では gedit にて説明します。security.conf を開き、表 2-1 のように編集し保存終了してください。

```
$ sudo gedit /etc/apache2/conf-available/security.conf
```

表 4 - 1 security.conf の変更箇所

行数	変更前	変更後
25	ServerTokens OS	ServerTokens Prod
36	ServerSignature On	ServerSignature Off

4.2.2. dir.conf

ディレクトリインデックスのファイル指定を変更するため、表 2-2 のように編集し保存終了します。

```
$ sudo gedit /etc/apache2/mods-available/dir.conf
```

表 4-2 dir.conf の変更箇所

行数	変更前	変更後
2	DirectoryIndex ¥ index.html index.cgi ¥ index.pl index.php ¥ index.xhtml	DirectoryIndex ¥ index.html index.php

4.2.3. apache2.conf

ServerName とタイムアウト時間(sec)の設定を行うために、apache2.conf を表 2-3 のように編集し保存終了します。

```
$ sudo gedit /etc/apache2/apache2.conf
```

表 4-3 apache2.conf の設定内容

行数	変更前	変更後
70	-	ServerName UoaServer
86	Timeout 300	Timeout 14400

4.2.4. 再起動

これまでの設定内容を反映させるために、Apache の再起動を行います。

```
$ sudo /etc/init.d/apache2 restart
```

5．Tomcat の導入

5.1．準備

Tomcat を実行するための専用ユーザーを作成します。

```
$ sudo useradd -M tomcat
```

5.2．インストール

(1) opt ホームディレクトリへ移動し Tomcat をダウンロードし、解凍してください。

```
$ cd /opt/
$ sudo curl -O https://archive.apache.org/dist/tomcat/tomcat-9/v9.0.4/bin/apache-tomcat-9.0.4.tar.gz
$ sudo tar -xzvf /opt/apache-tomcat-9.0.4.tar.gz
```

(2) 展開されたディレクトリの所有者を Tomcat ユーザーに変更します。

```
$ sudo chown -R tomcat:tomcat /opt/apache-tomcat-9.0.4
```

5.3．サービスファイル設定

(1) サービスとして起動させるために、service ファイル(ファイル名：tomcat.service)を新規作成します。記述内容は図 5-1 のとおりです。

```
$ sudo gedit /etc/systemd/system/tomcat.service
```

図 5 - 1 tomcat.service に記載する内容

```
[Unit]
 Description=Apache Tomcat 9
 After=network.target

 [Service]
 User=tomcat
 Group=tomcat
 Type=oneshot
 PIDFile=/opt/apache-tomcat-9.0.4/tomcat.pid
 RemainAfterExit=yes

 ExecStart=/opt/apache-tomcat-9.0.4/bin/startup.sh
 ExecStop=/opt/apache-tomcat-9.0.4/bin/shutdown.sh
 ExecReStart=/opt/apache-tomcat-9.0.4/bin/shutdown.sh;/opt/apache-tomcat-
```

```
9.0.4/bin/startup.sh

[Install]
WantedBy=multi-user.target
```

(2) service ファイル作成完了後、パーミッションを設定します。

```
$ sudo chmod 755 /etc/systemd/system/tomcat.service
```

(3) Tomcat を自動で起動させるための設定をします。

```
$ sudo systemctl enable tomcat
```

5.4. 起動

Tomcat を起動します。Web ブラウザから”http://localhost:8080/”にアクセスし、図 5-2 が出力されればインストールは完了です。

```
$ sudo service tomcat start
```

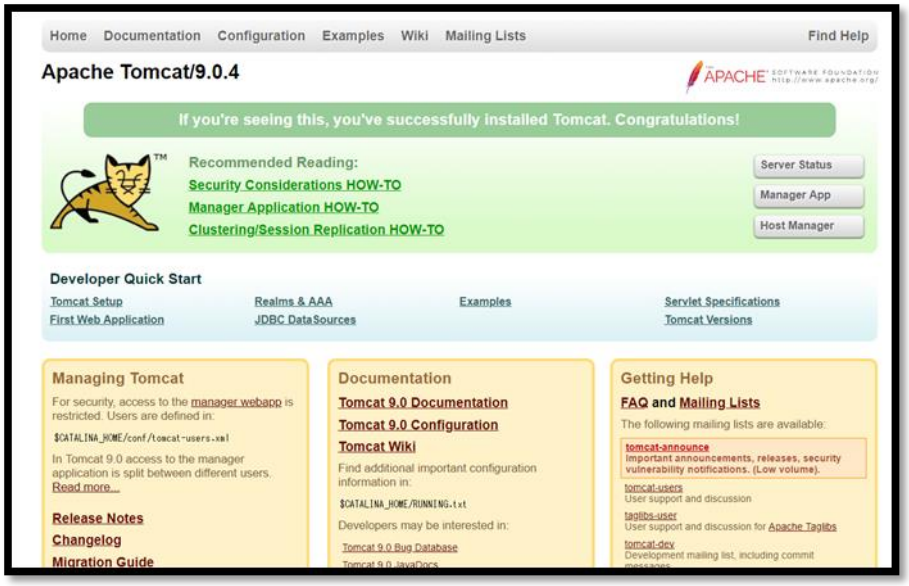

図 5 - 2 Tomcat 接続画面

6．Apache と Tomcat の連携

現状の設定では Apache と Tomcat が独立して動作するため、リクエストを Apache で待ち受けて Tomcat に渡すように設定を変更します。

6.1．プロキシ機能の有効化

プロキシ機能を組み込むため mod_proxy と mod_proxy_ajp のモジュールを有効にします。

```
$ sudo a2enmod proxy
$ sudo a2enmod proxy_ajp
```

6.2．仮想ホストの設定

(1) 仮想ホストの設定ファイルを新規作成します。ファイル名は任意ですが、本書では uoa_service.conf と命名し説明します。図 6-1 のように記述し保存終了します。

```
$ sudo gedit /etc/apache2/sites-available/uoa_service.conf
```

図 6 - 1 uoa_service.conf に記載する内容

```
<VirtualHost *:80>
        ServerName uoa_service.com
        ServerAdmin admin@uoa_service.com

        <Location />
            ProxyPass ajp://localhost:8009/
            Order allow,deny
            Allow from all
        </Location>
        ErrorLog /var/log/apache2/error.log

        # Possible values include: debug, info, notice, warn, error, crit,
        # alert, emerg.
        LogLevel warn

        CustomLog /var/log/apache2/access.log combined
</VirtualHost>
```

(2) uoa_service.conf のシンボリックを作成し仮想ホストを有効化します。その後、デフォルトで設定されているシンボリックを削除します。

```
$ sudo a2ensite uoa_service
$ sudo a2dissite 000-default.conf
```

6.3. Apache の再起動

これまでの設定内容を反映させるために、Apache2 を再起動します。

```
$ sudo /etc/init.d/apache2 restart
```

7．FFmpeg の導入

動画の最適化に必要なライブラリです。古い Version の FFmpeg がインストールされている場合はアンインストールしてください。

7.1．準備

(1) 事前に必要なライブラリをインストールします。

```
$ sudo apt install nasm
```

(2) ダウンロードしたソースコードを配置するディレクトリを作成します。

```
$ mkdir ~/src
```

7.2．Openh264 インストール

動画の最適化に必要な H.264/AVC のライブラリ「Openh264」をインストールします。ビルド時のコア数の指定は、CPU スペックに沿って指定ください。

```
$ cd ~/src
$ git clone https://github.com/cisco/openh264
$ cd openh264
$ git checkout v2.0.0 -b v2.0.0
$ make -j4
$ sudo make install
```

7.3．FFmpeg インストール

ビルド時のコア数の指定は、CPU スペックに沿って指定ください。

```
$ cd ~/src
$ git clone https://github.com/FFmpeg/FFmpeg
$ cd FFmpeg
$ git checkout n3.4.7 -b n3.4.7
$ ./configure --enable-gpl --enable-libopenh264 --enable-nonfree --enable-shared --enable-avresample --enable-avisynth
$ make -j4
$ sudo make install
```

7.4. 設定

(1) FFmpeg のライブラリを共通ライブラリに追加するため ld.so.conf に/usr/local/lib と/usr/lib を追記します。

```
$ sudo gedit /etc/ld.so.conf
```

図 7 - 1 ld.so.conf の記載内容

(2) キャッシュファイルを更新します。

```
$ sudo ldconfig
```

7.5. 起動確認

FFmpeg のインストール完了確認を行います。図 7-2 のように出力されれば完了です。

```
$ ffmpeg
```

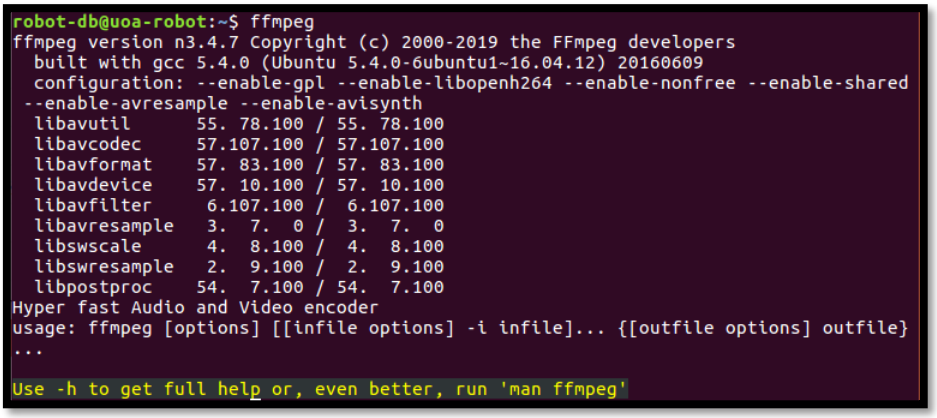

図 7 - 2 FFmpeg Version 確認結果画面

8．PostgreSQL の導入

ログインユーザー情報とアップロードする動画情報は DB に保存します。

8.1．インストール

(1) PostgreSQL 公式の apt リポジトリを/etc/apt/sources.list.d/pgdg.list に追加します。

```
$ sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ xenial-pgdg main" > /etc/apt/sources.list.d/pgdg.list'
```

(2) 公式鍵をインポートします。

```
$ wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -
```

(3) パッケージを更新します。

```
$ sudo apt-get update
```

(4) インストールをします。

```
$ sudo apt -y install postgresql-11
```

(5) サービスの自動起動を有効化します。

```
$ sudo systemctl enable postgresql
```

(6) サービスを起動します。

```
$ sudo systemctl start postgresql
```

(7) PostgreSQL シェルを起動します。以下のようなメッセージが表示されればインストール完了です。¥q で切断しますが、このあとに設定を行いますので、そのままにしてください。

```
$ sudo -u postgres psql
psql (11.4 (Ubuntu 11.4-1.pgdg16.04+1))
"help" でヘルプを表示します。
postgres=#
```

8.2. 設定

8.2.1. ロール作成

DB を管理するロールを作成します。便宜上、ユーザー名を db_user、パスワードを db_pass とします。

```
postgres=# CREATE ROLE db_user LOGIN CREATEDB PASSWORD 'db_pass';
```

8.2.2. DB 作成

DB を作成します。便宜上、DB 名は db_name、オーナーは db_user とします。

```
postgres=# CREATE DATABASE db_name OWNER db_user;
```

8.2.3. テーブルとシーケンス作成

(1) データを格納するテーブルとシーケンスを作成しますので、(9)で作成した DB に切り替えます。

```
postgres=#¥c db_name
```

(2) 動画情報を格納するテーブル video_tbl を作成します。

```
create table "public".video_tbl (
video_id integer not null
, robot_id character varying(256)
, file_path character varying(512)
, file_name character varying(512)
, shooting_location character varying(256)
, shooting_datetime timestamp
, memo_label character varying(512)
, meta_file_path character varying(512)
, primary key (video_id)
);
```

(3) ユーザー情報を格納するテーブル user_tbl を作成します。

```
create table "public".user_tbl (
user_id integer not null
, user_name character varying(256)
, password character varying(256)
, primary key (user_id)
);
```

(4) ビデオ ID シーケンス video_id_seq を作成します。

```
create sequence video_id_seq
        increment by 1
        maxvalue 99999999
        start with 1
        no cycle;
```

8.2.4. ユーザー登録

ログインユーザー情報を登録します。本書では予め用意したユーザー情報(CSV 形式)をインポートして登録することとします。

```
COPY user_tbl (user_id, user_name, password) from
'/home/(ユーザー名)/user_tbl.csv' with csv;
```

8.2.5. オーナー変更

作成したテーブルとシーケンスのオーナーを変更します。

```
ALTER TABLE video_tbl OWNER TO db_user;
ALTER TABLE user_tbl OWNER TO db_user;
ALTER SEQUENCE video_id_seq OWNER TO db_user;
```

9．Spring の導入

DB の接続設定ファイルの編集や War ファイルの生成に必要な Spring Tools をインストールします。

2020/2/18 現在、Spring Tools 3.9.9 のダウンロードページならびにファイルの提供を停止したようですので、最新版(4 系)をダウンロードしてください。本システムは 4 系での動作検証は行っていませんが、簡易確認はしております。本書では 3.9.9 を導入していますが、4 系の場合は置き換えてお読みください。

9.1．準備

作業スペースを作成します。本書ではホームディレクトリに Work ディレクトリを作成して導入手順を説明することにします。

```
$ mkdir ~/Work && cd ~/Work
```

9.2．ダウンロード

(1) 3 系の最新版である 3.9.9 を以下からダウンロードします。今回はサイトからダウンロードしましたが、ターミナル上で wget コマンドによるダウンロードも可能です。サイトからダウンロードする場合は、ダウンロード完了後に Work ディレクトリにファイルを移動してください。
https://download.springsource.com/release/STS/3.9.9.RELEASE/dist/e4.12/spring-tool-suite-3.9.9.RELEASE-e4.12.0-linux-gtk-x86_64.tar.gz

(2) 解凍します。

```
$ tar -xzvf ~/Work/spring-tool-suite-3.9.9.RELEASE-e4.12.0-linux-gtk-x86_64.tar.gz
```

9.3．日本語化

(1) Eclipse 版をダウンロードしましたので、以下から日本語化プラグイン(Pleiades)の Linux 版を導入します。https://mergedoc.osdn.jp/

(2) 解凍します。

```
$ unzip ~/Work/pleiades.zip -d pleiades
```

(3) 展開された pleiades の中にある" features"内の" jp.sourceforge.mergedoc.pleiades"と"plugins"内の" jp.sourceforge.mergedoc.pleiades"ディレクトリを展開された Spring Tool Suite にある"features"と"plugins"にコピーします。

```
$ cp -r ~/Work/pleiades/plugins/jp.sourceforge.mergedoc.pleiades ~/Work/sts-bundle/sts-3.9.9.RELEASE/plugins/.
$ cp -r ~/Work/pleiades/features/jp.sourceforge.mergedoc.pleiades ~/Work/sts-bundle/sts-3.9.9.RELEASE/features/.
```

(4) 設定ファイル(STS.ini)を編集します。ファイルを開くと 9 行目に” -vmargs”と記載があり、改行し以下の内容を追記します。

```
-Xverify:none
-javaagent:plugins/jp.sourceforge.mergedoc.pleiades/pleiades.jar
```

図 9 - 1 追記内容

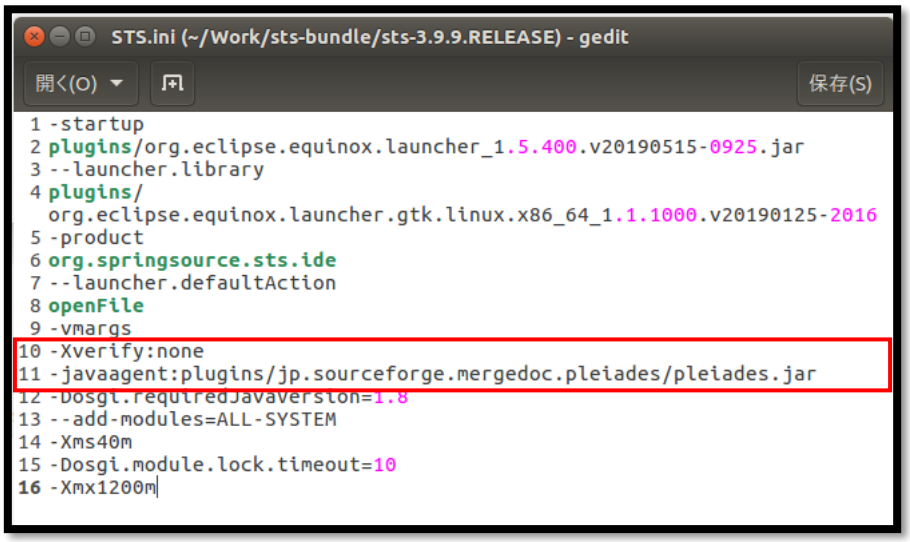

図 9 - 2 追記完了後の STS.ini

9.4. Maven

9.4.1. インストール

Java のビルドツールである Maven をインストールします。

```
$ sudo apt install maven
```

9.4.2. 動作確認

バージョンを確認し、図 9-3 のとおりであれば、インストール完了です。

```
$ mvn -v
```

図 9 - 3 maven バージョン情報

9.5. 設定

展開された sis-bundle 内の sts-3.9.9.RELEASE にある STS をコマンド実行またはダブルクリックで Spring Tool Suite が起動します。

9.5.1. Maven

9.5.1.1. インストール

(1) Spring Tool Suite を起動後、インストールした Maven を適用するための設定をおこないます。ツールバーの[ウィンドウ]から設定を開きます。

(2) Maven の中にあるインストールを選択し、”Maven の起動で使用するインストールの選択”にある”追加”を押下します。

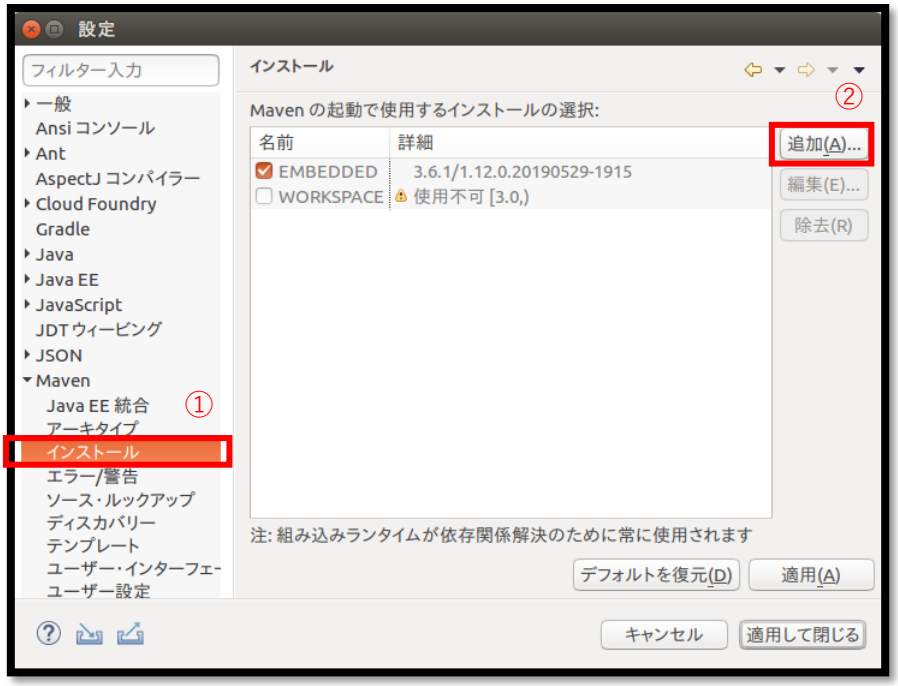

図 9-4 Maven インストール設定画面

(3) 新規 Maven ランタイムウィンドウの“ディレクトリー...”ボタンを押下し、インストールした Maven のパスをインストールホーム設定します。パスは/usr/share/maven です。記述したら、完了ボタンを押下します。

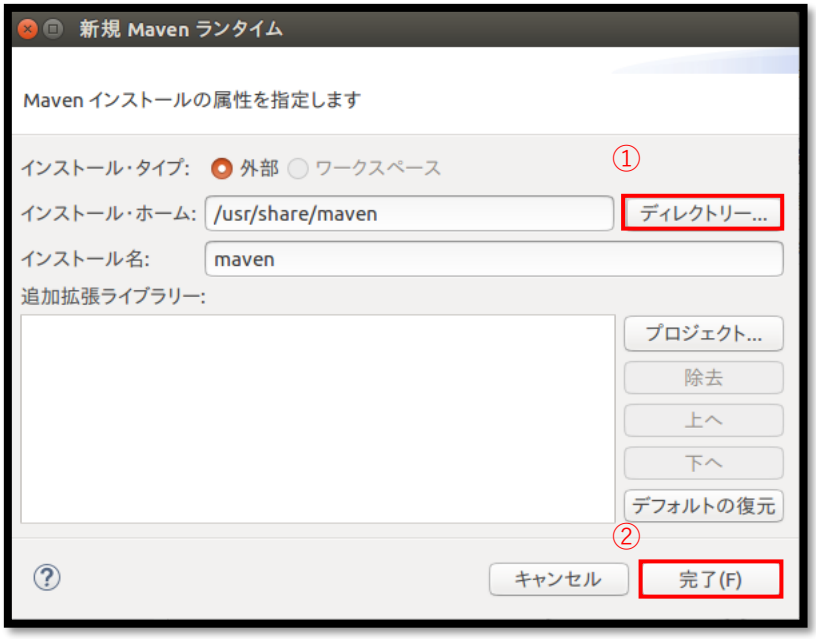

図 9 - 5 インストールパスの設定

(4) 追加された maven を選択し、”適用”を押下すると設定完了です。

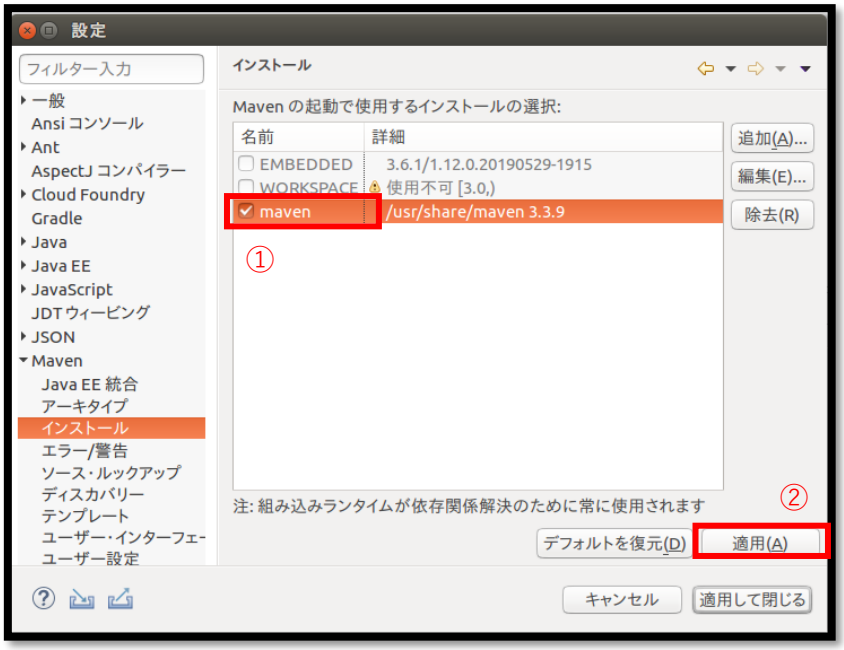

図 9 - 6 インストールの選択

9.5.1.2. ユーザー設定

プロキシ環境下でインターネット接続している場合は Maven のユーザー設定を変更する必要があります。

(1) Maven をインストールした際に設定ファイルを格納されているディレクトリ(/etc/maven)にある settings.xml をリネームしバックアップします。

```
$ sudo mv /etc/maven/settings.xml /etc/maven/settings_bk.xml
```

(2) バックアップ後、settings.xml を作成します。

```
$ sudo gedit /etc/maven/settings.xml
```

(3) 記述内容を図 9-7 に示します。プロキシ情報は<host>と<port>に記述します。記述完了したら保存終了します。

```
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
                      http://maven.apache.org/xsd/settings-1.0.0.xsd">
<proxies>
    <proxy>
      <id>optional</id>
      <active>true</active>
      <protocol>https</protocol>
      <host>your.proxy.address</host>
      <port>****</port>
    </proxy>
</proxies>
</settings>
```

図 9 - 7 setting.xml 記述内容

(4) Spring Tool Suite の設定を開き、Maven にあるユーザー設定を選択します。ユーザー設定にあるグローバル設定およびユーザー設定の参照からそれぞれ先ほどの settings.xml を選択し、適用を押下することで変更内容を反映します。[図 9-8]

図 9 - 8 Maven ユーザー設定画面

9.6. プロジェクトファイルのインポート

(1) ツールバーの[ファイル]からインポートを選択します。

(2) インポート画面が表示されますので、「一般」ディレクトリから「既存プロジェクトをワークスペースへ」を選択し、”次へ”を押下します。

(3) “アーカイブ・ファイルの選択”を選択し、”参照”からプロジェクトを指定します。本書では VDBS プロジェクトと命名されたプロジェクトをインポートします。ファイルを選択後、”完了”を押下するとプロジェクトがインポートされます。

図 9-9 インポート完了

9.7. プロジェクトの更新

(1) インポートしたプロジェクトを右クリックし、[Maven]-[プロジェクトの更新]を選択します。

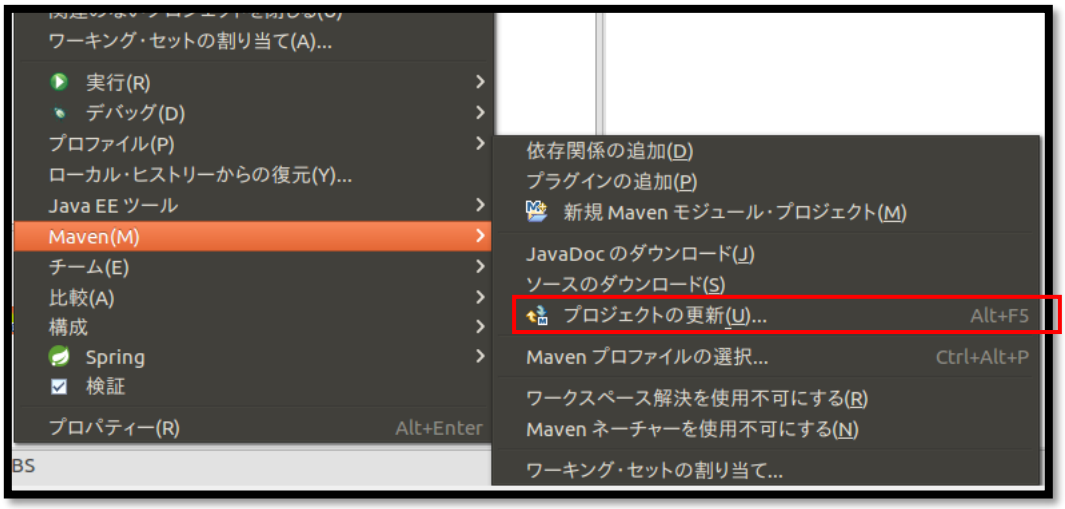

図 9-10 [Maven]-[プロジェクトの更新]を選択

(2) プロジェクトにチェックがされていることを確認し、OK を押下します。

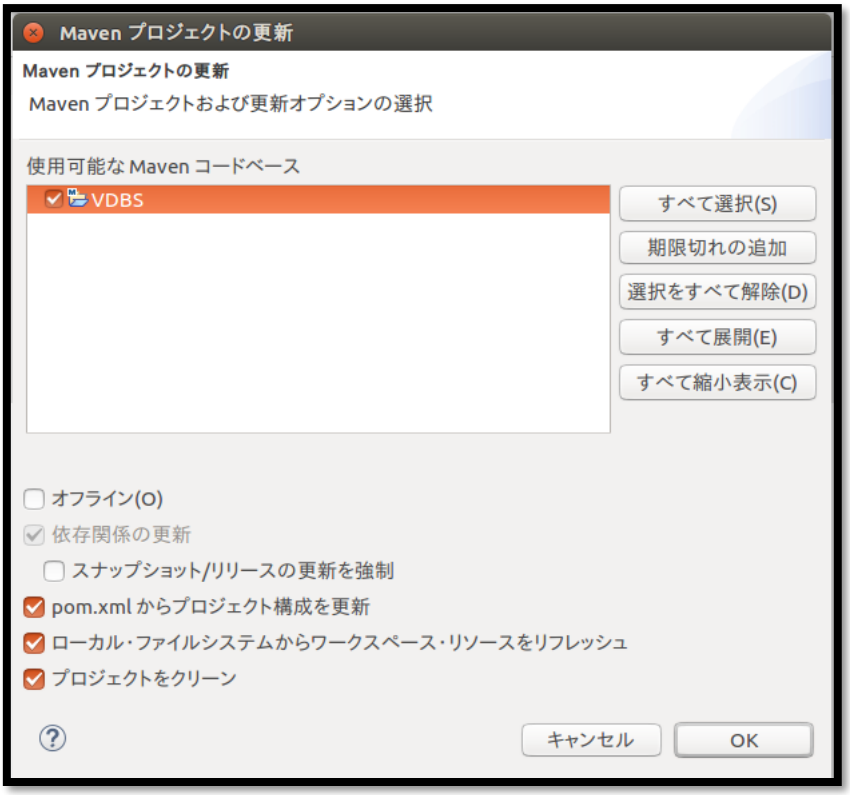

図 9-11 Maven プロジェクトの更新

9.8. database.properties の設定

Spring Tool Suite プロジェクトには DB 接続設定用ファイルがあり、DB やユーザーなどに応じて変更が必要になります。

(1) パッケージ・エクスプローラから VDBS プロジェクトの/src/main/resource にある database.properties を選択して開きます。

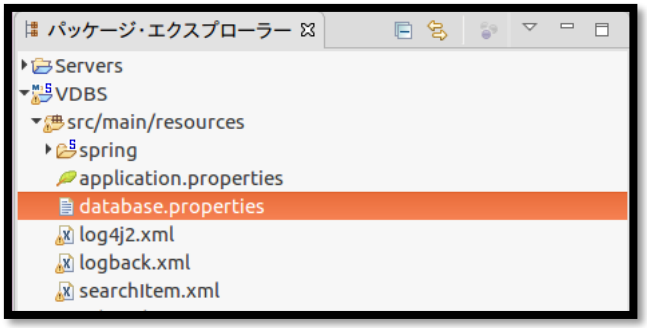

図 9-12 プロジェクトの database.properties を選択

(2) database.properties に「8.PostgreSQL の導入」で作成したユーザー、パスワード、DB 名にそれぞれ記述します。

図 9-13 database.properties

表 9-1 database.properties の説明

項目	詳細
jdbc.driverClassName	DB へアクセスするためのライブラリを設定 変更不要
jdbc.url	接続先の DB への URL ”jdbc:postgresql://localhost:5432/ “の変更は不要で、 DB 名のみ変更
jdbc.username	DB に接続するユーザーを設定
jdbc.password	DB に接続する際のパスワードを設定

9.9. War ファイルの生成

Tomcat にデプロイする War ファイルを Spring Tool Suite で生成します。

9.9.1. ターゲットディレクトリのクリーン

target ディレクトリに生成されているファイルを削除します。VDBS プロジェクトを右クリックし、実行から Maven clean を実施します。コンソールに完了が通知されると以前の Maven install で生成されたファイルが削除されます。

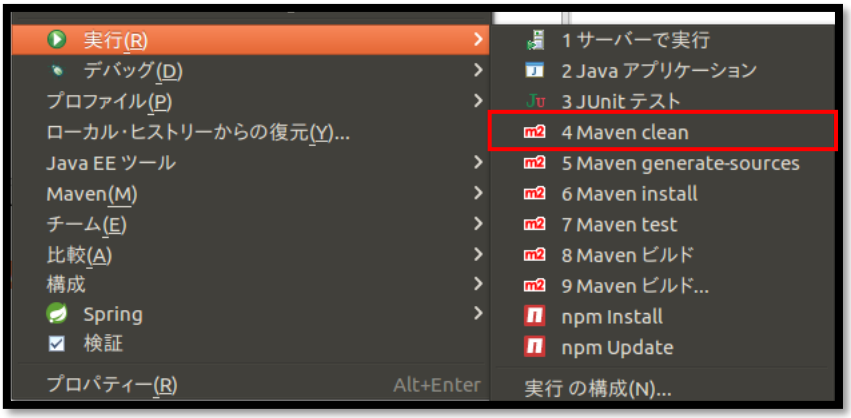

図 9-14 Maven clean の選択

9.9.2. War ファイルの生成

プロジェクトを右クリックし、実行から Maven install を実施します。コンソールに完了が通知されるとプロジェクト内の target ディレクトリに War ファイルが生成されます。

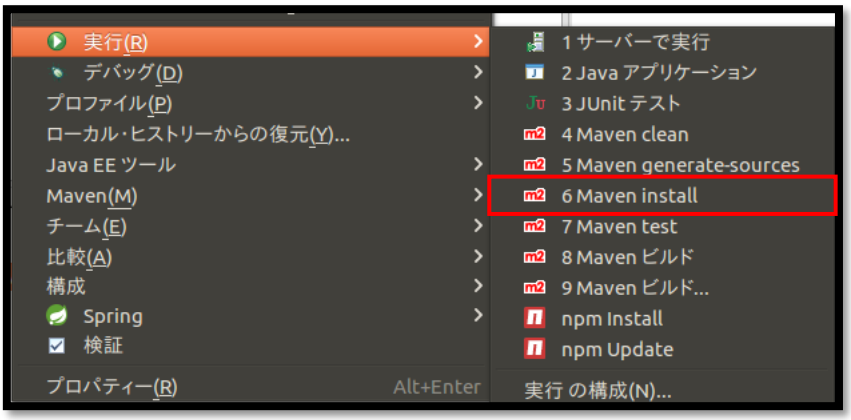

図 9-15 Maven install の選択

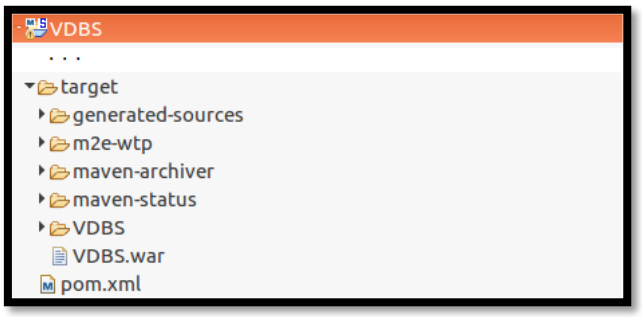

図　9 - 1 6　生成された War ファイル

１０．Web アプリケーションの設定

システムに必要な環境構築が完了しましたら web アプリケーションの設定を実施します。本システムは図 10-1 の構成とします。

```
/opt/
└── vdb/
    ├── logs/          ←出力されたログを格納
    └── data/
        ├── upload/    ← アップロードされた動画(オリジナル)を格納
        ├── video/     ← オリジナルを基に最適化した動画を格納
        └── download/  ← ダウンロード用の Zip ファイルを格納
```

図 １０-１ 動画 DB システム ディレクトリ構成

１０.１．Web アプリケーション用ディレクトリ作成

システムに必要なログ出力用ディレクトリ、データ格納ディレクトリ等のディレクトリを作成し、パーミッションを設定します。

```
$ sudo mkdir /opt/vdb
$ sudo mkdir /opt/vdb/logs
$ sudo mkdir /opt/vdb/data
$ sudo mkdir /opt/vdb/data/video
$ sudo mkdir /opt/vdb/data/upload
$ sudo mkdir /opt/vdb/data/download
$ sudo chmod -R 755 /opt/vdb/*
```

１０.２． ディレクトリ所有者設定

vdb ディレクトリの所有者を Tomcat 専用ユーザーに変更します。

```
$ sudo chown -R tomcat:tomcat /opt/vdb
```

１０.３． Tomcat シンボリックリンク参照有効化

Tomcat のシンボリックリンク参照設定を有効にします。context.xml ファイルを開き <Context> タグのすぐ後に以下を追記して保存終了します。

```
$ sudo gedit /opt/apache-tomcat-9.0.4/conf/context.xml
```

表 １０-１ 追記する文章

<Resources className="org.apache.catalina.webresources.StandardRoot" allowLinking="true"/>

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--
3   Licensed to the Apache Software Foundation (ASF) under one or more
4   contributor license agreements.  See the NOTICE file distributed with
5   this work for additional information regarding copyright ownership.
6   The ASF licenses this file to You under the Apache License, Version 2.0
7   (the "License"); you may not use this file except in compliance with
8   the License.  You may obtain a copy of the License at
9
10      http://www.apache.org/licenses/LICENSE-2.0
11
12  Unless required by applicable law or agreed to in writing, software
13  distributed under the License is distributed on an "AS IS" BASIS,
14  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
15  See the License for the specific language governing permissions and
16  limitations under the License.
17 -->
18 <!-- The contents of this file will be loaded for each web application -->
19 <Context>
20 <Resources className="org.apache.catalina.webresources.StandardRoot" allowLinking="true"/>
21     <!-- Default set of monitored resources. If one of these changes, the    -->
22     <!-- web application will be reloaded.                                   -->
23     <WatchedResource>WEB-INF/web.xml</WatchedResource>
24     <WatchedResource>WEB-INF/tomcat-web.xml</WatchedResource>
25     <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>
```

図 １０-２ context.xml の変更箇所

１１．Web アプリのデプロイ

１１.１．war ファイルの設定

(1) Tomcat のサービスを停止します。

```
$ sudo service tomcat stop
```

(2) VDBS.war ファイルを Tomcat の webapps ディレクトリにコピーし、パーミッションの設定をします。本書では便宜上ユーザーディレクトリに配置されているものとします。

```
$ cd ~
$ sudo cp VDBS.war /opt/apache-tomcat-9.0.4/webapps/.
$ sudo chmod 755 /opt/apache-tomcat-9.0.4/webapps/VDBS.war
```

(3) Tomcat のサービスを開始します。

```
$ sudo service tomcat start
```

(4) Tomcat のユーザーで war ファイルが展開されていることを確認します。（VDBS ディレクトリの更新日時で確認してください。）[図 11-1]

```
$ sudo su - tomcat
$ ls -l /opt/apache-tomcat-9.0.4/webapps/VDBS
```

```
$ ls -l /opt/apache-tomcat-9.0.4/webapps/VDBS
合計 12
drwxr-x--- 3 tomcat tomcat 4096  2月  7 18:14 META-INF
drwxr-x--- 5 tomcat tomcat 4096  2月  7 18:14 WEB-INF
drwxr-x--- 8 tomcat tomcat 4096  2月  7 18:14 resources
```

図 １１-１ 更新日時確認時 画面

１１.２．Tomcat サービス停止

Tomcat のサービスを停止します。

```
$ sudo service tomcat stop
```

１１.３． シンボリックリンクの作成

Tomcat のユーザーでシンボリックリンクを作成します。

```
$ ln -s /opt/vdb/data /opt/apache-tomcat-9.0.4/webapps/VDBS/resources/data
```

１１.４． 共有ライブラリの設定

以下の共有ライブラリを Tomcat ディレクトリにコピーします。本書では便宜上ユーザーディレクトリに配置されているものとします。コピー完了後に、パーミッションの設定をします。

・libVDBSVideoQualityOptimizer.so

```
$ cd ~
$ sudo cp libVDBSVideoQualityOptimizer.so /opt/apache-tomcat-9.0.4/webapps/VDBS/WEB-INF/classes/.
$ sudo chmod 755 /opt/apache-tomcat-9.0.4/webapps/VDBS/WEB-INF/classes/libVDBSVideoQualityOptimizer.so
```

１１.５． Tomcat サービス開始

Tomcat のサービスを開始し、アプリケーションを開始させます。

```
$ sudo service tomcat start
```

著作権

本文書の著作権は公立大学法人 会津大学に帰属します。
この文書のライセンスは以下のとおりです。
クリエイティブ・コモンズ 表示 2．1日本
http://creativecommons.org/licenses/by/2.1/jp/